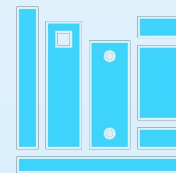




App Service



SIGNAL R

Proyecto de Investigación

SIGNAL R

Es una biblioteca para ASP. NET usada principalmente para adaptar funcionalidades web en *tiempo real*.

- Es de código abierto.
- El servidor inserta contenido en los clientes conectados a medida que esté disponible, en lugar de tener que esperar a que un cliente solicite nuevos datos.
- Proporciona una API sencilla para crear llamadas a procedimientos remotos (RPC)* de servidor a cliente.
- Utiliza técnicas de TRANSPORTE y CONCENTRADORES para controlar conexiones automáticamente y enviar mensajes a todos de manera simultanea.
- La conexión entre el cliente y el servidor es persistente, a diferencia de una conexión HTTP clásica, que se vuelve a establecer para cada comunicación.

*RPC: Interacción remota en otro equipo de la red, simulando un procedimiento local



TRANSPORTE



- Para el envío de actualizaciones en tiempo real desde el servidor al cliente utiliza varias técnicas o mecanismos de comunicación. Esta depende de la disponibilidad y compatibilidad del cliente y del servidor. Elige automáticamente el mejor método de transporte que se encuentra dentro de las funcionalidades del servidor y el cliente

Algunos transportes son:

- WEBSOCKETS
- Eventos enviados por el servidor
- Sondeos Largos
- Long Polling

TRANSPORTE



WEBSOCKETS

Tecnología de comunicación bidireccional que permite a los servidores enviar información a los clientes de forma asíncrona. Posee una latencia baja y solo se admite en los navegadores mas recientes, es el mas estricto. Siempre que sea posible y esté disponible SignalR optará por esta opción.

Eventos Enviados Por El Servidor (SSE)

SignalR utiliza esta tecnología cuando los WebSockets no están disponibles, ya que es más eficiente que el Long Polling.

A través de esta técnica un servidor puede enviar actualizaciones en tiempo real a un cliente, sin que este tenga que solicitarlas continuamente.

TRANSPORTE



Long Polling

Esta técnica utiliza una conexión HTTP que se mantiene abierta durante un tiempo determinado para recibir actualizaciones en tiempo real. SignalR utiliza Long Polling cuando ni WebSockets ni SSE están disponibles en el cliente o en el servidor.

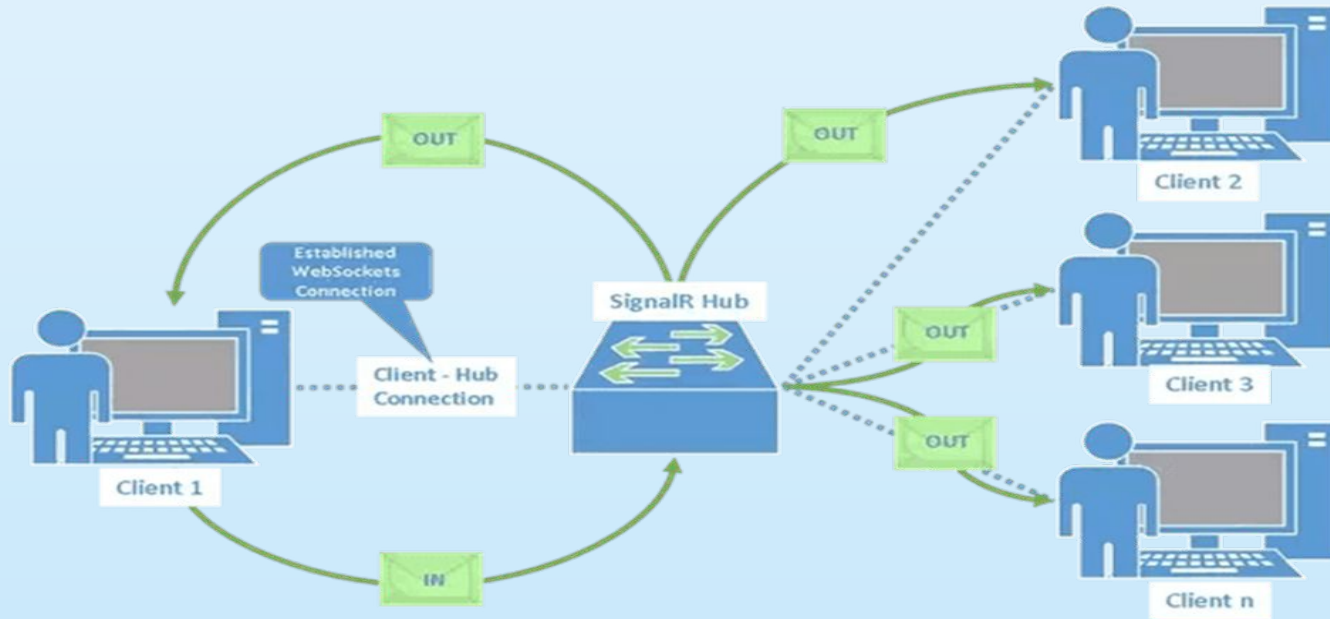
Sondeos Largos

Solo es compatible con Internet Explorer, y utiliza un IFrame oculto y una conexión unidireccional para transmitir datos desde el servidor al cliente. Ha sido reemplazado por otras tecnologías más modernas y eficientes, debido a su incompatibilidad con otros navegadores y a su ineficiencia en términos de uso de recursos del servidor.

CONCENTRADORES / HUBS

- Es una clase que actúa como intermediario para la comunicación en tiempo real entre el servidor y los clientes conectados.
- Permite a los servidores y clientes conectados definir e invocar los métodos entre sí.
- El servidor define los métodos que se llaman desde el cliente y el cliente define los métodos que se llaman desde el servidor.

CONCENTRADORES / HUBS



CONCENTRADORES / HUBS

Lado del Servidor

```
using Microsoft.AspNetCore.SignalR;

public class ChatHub : Hub
{
    public async Task SendMessage(string user, string message)
    {
        await Clients.All.SendAsync("ReceiveMessage", user, message);
    }
}
```


CONCENTRADORES / HUBS – Propiedades

Lado del Servidor

- **Clients**

- All
- Caller
- Group("Chat Users")
- Others

...

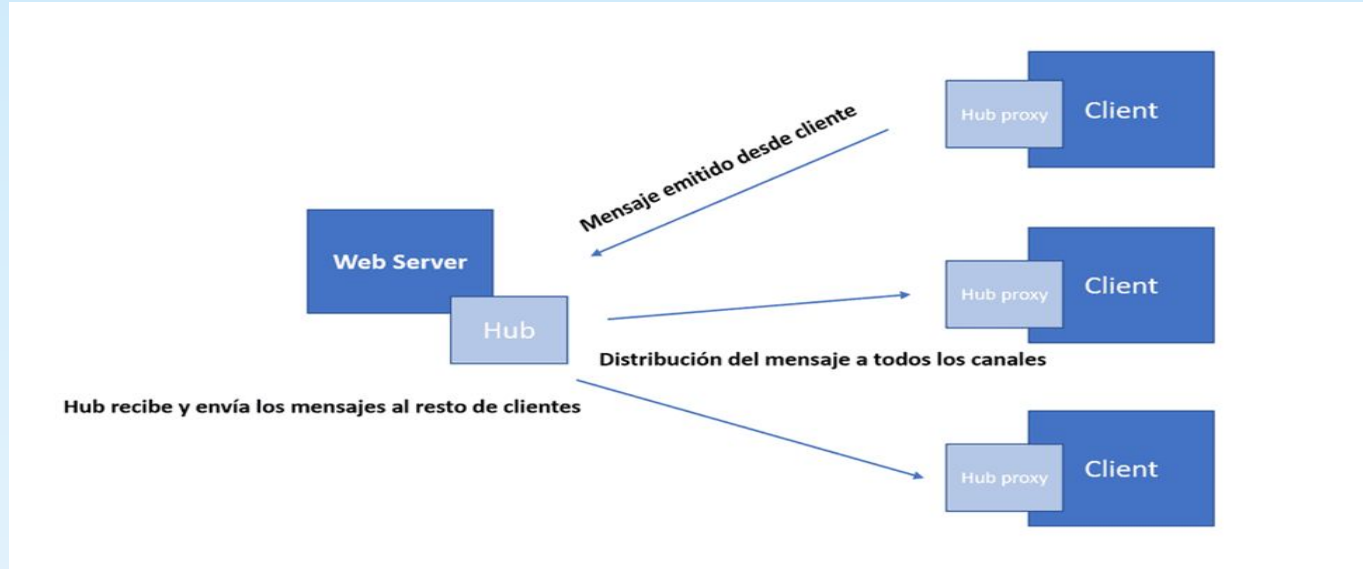
- **Context**

- ConnectionId
- UserIdentifier
- ConnectionAborted

...

CONCENTRADORES / HUBS

Lado del Cliente



CONCENTRADORES / HUBS

Lado del Cliente

```
const connection = new signalR.HubConnectionBuilder()  
    .withUrl("/chatHub")  
    .build();
```

```
connection.on("ReceiveMessage", function (user, message) {  
    // Lógica para mostrar el mensaje recibido en el cliente  
    console.log(`${user}: ${message}`);  
});
```

Casos de Uso



Chat en Vivo



Transmisión de Datos



Colaboración Multijugador



Notificaciones



Seguimiento de Ubicación GPS

Azure SignalR Service



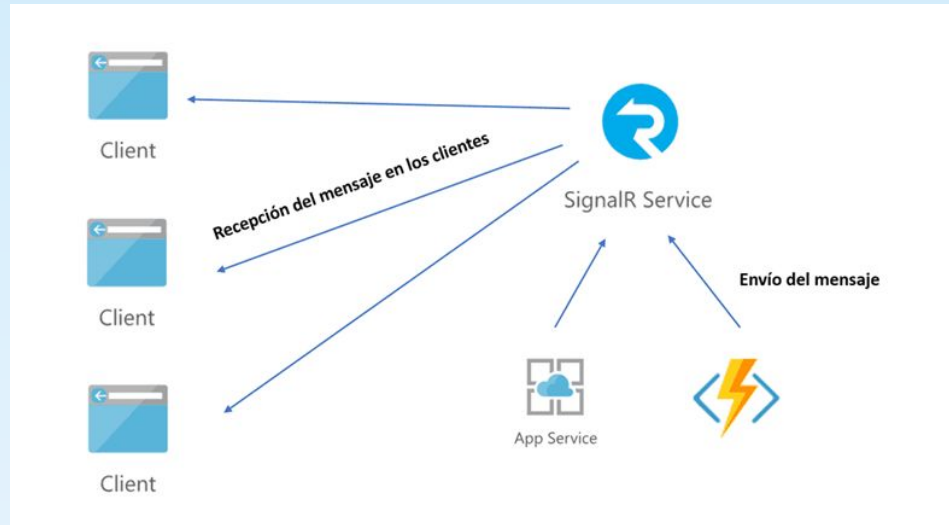
En el caso de tener un trafico excesivo en nuestro concentrador(hub), o que la arquitectura no responda adecuadamente.

Podemos tener un servicio único y escalable

Azure SignalR Service

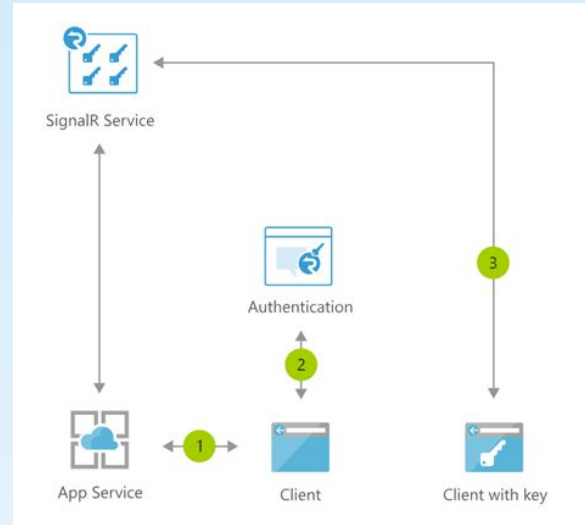
Permitiendo que nuestro backend de SignalR, este alojado en Azure.
Lo que haría que nuestro concentrador(hub), ya no se ejecutara dede la web.

Podremos conectar a un servicio tanto clientes como digamos, pudiendo escalar el servicio bajo demanda.



Azure SignalR Service

Las conexiones al servicio se realizan con autenticación multifactor, gracias a la clave de acceso, que nos devuelve un token de sesión válido para que las aplicaciones clientes consuman el servicio.



Preguntas??

